📘Project Finalization Document

AI-Powered Multilingual Video Dubbing Application

Status: FINAL & LOCKED – Ready for Development
Purpose: Resume / Portfolio Project to demonstrate real-world software engineering skills
Methodology: Scrum
Target Users: 12–20 approved users only

---

1. Project Overview

This project is a **native mobile application** that allows approved users to upload short videos and receive a **dubbed version** of the video in a user-selected target language.

The system performs: 1. Speech-to-Text (caption extraction) 2. Translation to the requested language 3. Subtitle generation 4. Text-to-Speech audio synthesis 5. Final video + audio merge

The application is intentionally **cost-controlled, legally safe, and limited in scope**, designed to run entirely on **free-tier and open-source infrastructure**.

---

2. Core Objectives (Why This Project Exists)

- Demonstrate **end-to-end system design**
- Showcase **backend architecture & API design**
- Highlight **database modeling (MongoDB)**
- Show **AI pipeline integration** (STT, translation, TTS)
- Apply **Scrum methodology** realistically
- Prove ability to make **engineering trade-offs**

This project is **not a business product** and is **not intended for mass public usage**.

---

3. Key Constraints (Intentionally Applied)

Constraint	Value	Reason
Active users	Max 20	Cost & abuse control
Video length	5–10 seconds	CPU & storage control
Video source	User-uploaded only	Legal safety
Storage duration	3 hours	Cost + privacy

Constraint	Value	Reason
User access	Approved only	Predictable usage
Platform	React Native	Native skills showcase

These constraints are **design features**, not limitations.

4. Technology Stack (Final)

Frontend

- **React Native** (Expo)
- REST API communication
- JWT-based authentication

Backend

- **Node.js / FastAPI** (implementation choice flexible)
- FFmpeg for media processing
- Async/background job handling

Database

- **MongoDB Atlas (Free Tier – M0)**
- Stores users, video metadata, and job states
- Uses **TTL indexes** for automatic cleanup

Media Storage

- **Cloudinary (Free Tier)**
- Stores original & processed videos temporarily
- Videos deleted after 3 hours

AI Components (Open-Source)

- Speech-to-Text: Whisper (small/base)
- Translation: Marian / NLLB
- Text-to-Speech: Coqui TTS

5. User Access Model

Access Policy

- Application is **invite-only / admin-approved**
- Unapproved users cannot process videos

Roles

- **Admin:** Approves users
- **User:** Uploads videos and downloads results

This ensures: - Zero abuse - Predictable compute usage - Sustainable free deployment

---

6. Data Model (MongoDB)

Users Collection

- email
- role (user / admin)
- approved (boolean)
- createdAt

Videos Collection

- userId (reference)
- originalVideoUrl
- processedVideoUrl
- sourceLanguage
- targetLanguage
- status (processing / completed / failed)
- expiresAt
- createdAt

Jobs Collection (Optional, Advanced)

- videoId
- processingStep
- status
- timestamps

---

7. Data Lifecycle & Cleanup

- Each processed video has an `expiresAt` timestamp
- MongoDB **TTL index** auto-deletes expired records
- Cloudinary assets are deleted via backend cleanup
- Users are warned: **download within 3 hours**

No long-term data retention.

---

8. Cost Strategy (Zero-Cost Guarantee)

Area	Strategy
Storage	Cloudinary free tier + auto deletion
Compute	CPU-only + short videos
Database	MongoDB Atlas M0
Auth	Custom JWT
Hosting	Free-tier backend

With max 20 users, the system **stays within free limits indefinitely**.

---

9. Legal & Compliance Considerations

- Only user-uploaded content accepted
- No third-party video downloading
- Temporary storage only
- No redistribution beyond user download
- Explicit user warning about deletion

Designed to minimize copyright and privacy risks.

---

10. Scalability Philosophy

This system is **intentionally not horizontally scalable**.

Future scalability is discussed conceptually but not implemented: - Queue systems - GPU inference - Object storage

This demonstrates **engineering judgment**, not under-engineering.

---

11. Scrum Execution Plan

Sprint 1

- Auth (JWT)
- Approved user flow
- Video upload API

Sprint 2

- AI processing pipeline
- MongoDB integration
- Job tracking

Sprint 3

- Cloudinary integration
- Cleanup logic
- Error handling

Sprint 4

- UI polish
- Documentation
- Demo readiness

---

12. Success Criteria

- App works reliably for approved users
- Short video processed end-to-end
- Automatic deletion works
- Clean codebase & documentation
- Strong explanation during interviews

---

13. Final Statement

This project is **finalized and locked**.

All technical, legal, and cost constraints have been consciously addressed. The system is now **ready for implementation**.

> The goal is not scale — the goal is engineering clarity.